

图文 109 当我们发送消息的时候，是如何从NameServer拉取Topic元数据的？

231 人次阅读

2020-03-09 15:15:28

[详情](#) [评论](#)

当我们发送消息的时候，是如何从NameServer拉取Topic元数据的？



继《从零开始带你成为JVM实战高手》后，阿里资深技术专家携新作再度出山，重磅推荐：

(点击下方蓝字试听)

[《从零开始带你成为MySQL实战优化高手》](#)

之前我们已经给大家讲到了发送消息到Broker的时候，使用的是Producer来发送，也大概介绍了一下Producer初始化的过程

其实初始化的过程极端的复杂，但是我们却真的不用过于的深究，因为其实比如拉取Topic的路由数据，选择MessageQueue，跟Broker构建长连接，发送消息过去，这些核心的逻辑，都是封装在发送消息的方法中的。

因此我们今天就从发送消息的方法开始讲起，实际上当你调用Producer的send()方法发送消息的时候，这个方法调用会一直到比较底层的逻辑里去，最终会调用到DefaultMQProducerImpl类的sendDefaultImpl()方法里去，在这个方

法里，上来什么都没干，直接就有一行非常关键的代码，如下。

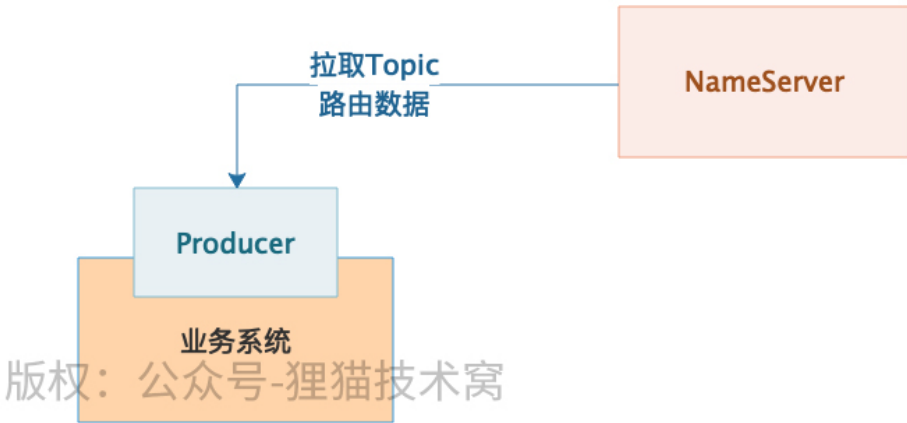
```
TopicPublishInfo topicPublishInfo = this.tryToFindTopicPublishInfo(msg.getTopic());
```

其实看到这行代码，大家就什么都明白了，每次你发送消息的时候，他都会先去检查一下，这个你要发送消息的那个Topic的路由数据是否在你客户端本地

如果不在的话，必然会发送请求到NameServer那里去拉取一下的，然后缓存在客户端本地。

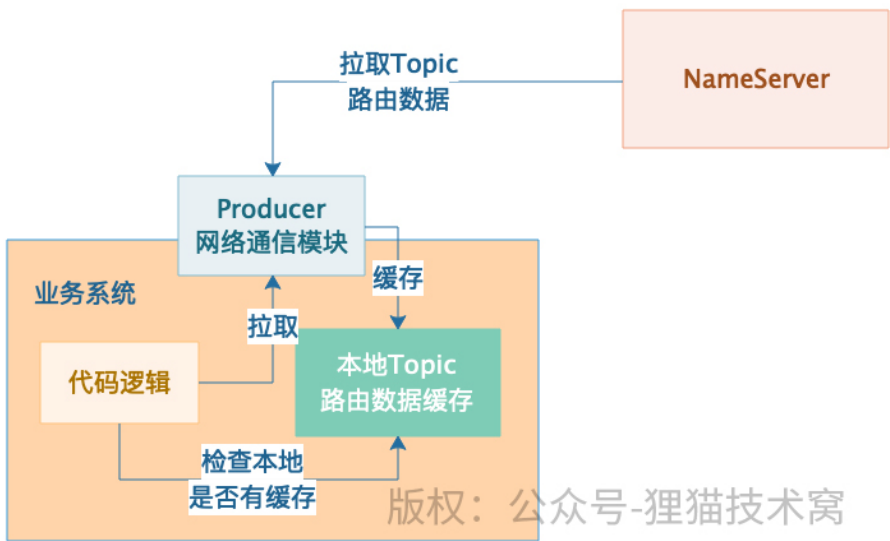
所以今天我们就重点来看看，这个Producer客户端运行在你的业务系统里的时候，他如何从NameServer拉取到你的Topic的路由数据的？

我们看下图的一个简单示意



其实当你进入了this.tryToFindTopicPublishInfo(msg.getTopic())这个方法逻辑之后，会发现他的逻辑非常的简单

其实简单来说，他就是先检查了一下自己本地是否有这个Topic的路由数据的缓存，如果没有的话就发送网络请求到NameServer去拉取，如果有的话，就直接返回本地Topic路由数据缓存了，如下图的逻辑演示。



具体的一些源码细节，我们就不给大家贴出来了，其实看源码，一个是看源码的技巧，一个就是从源码里提取核心业务逻辑和流程，之前我们已经给大家讲了很多看源码的技巧了

大家只要按我们的思路去看，都能大致看懂源码，现在开始，更重要的是，我们使用狸猫技术窝的专栏特有的风格，就是一步一图的方式，用图给大家把源码的流程讲清楚！

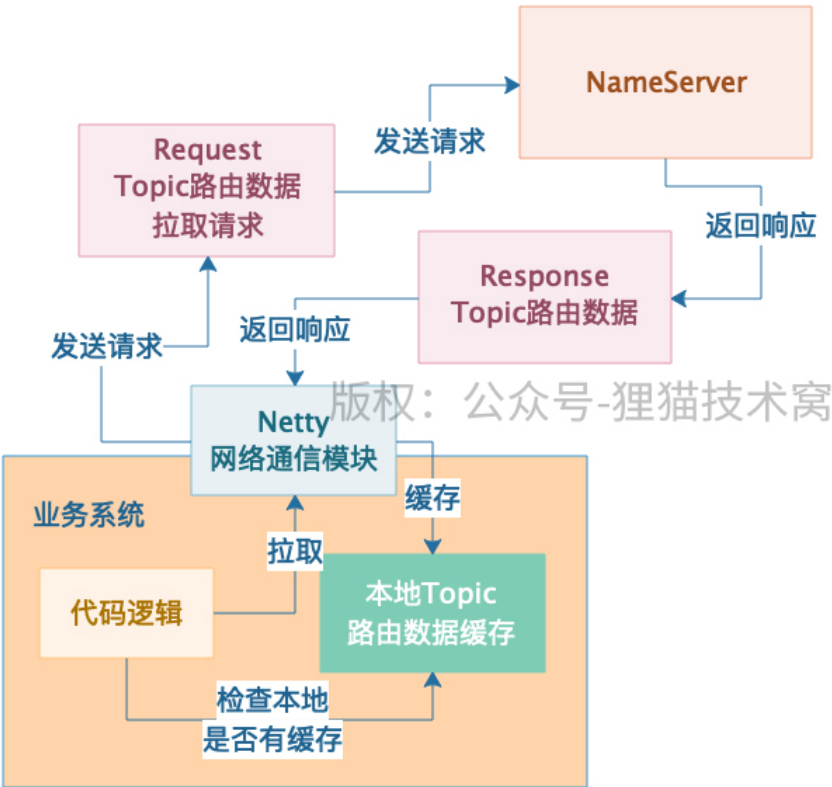
所以接着我们当然很想知道的是，Producer到底是如何发送网络请求到NameServer去拉取Topic路由数据的，其实这里就对应了tryToFindTopicPublishInfo()方法内的一行代码，我们看看。

```
this.mQClientFactory.updateTopicRouteInfoFromNameServer(topic);
```

通过这行代码，他就可以去从NameServer拉取某个Topic的路由数据，然后更新到自己本地的缓存里去了。

具体的发送请求到NameServer的拉取过程，其实之前都大致讲解到了，简单来说，就是封装一个Request请求对象，然后通过底层的Netty客户端发送请求到NameServer，接收到一个Response响应对象。

然后他就会从Response响应对象里取出来自己需要的Topic路由数据，更新到自己本地缓存里去，更新的时候会做一些判断，比如Topic路由数据是否有改变过，等等，然后把Topic路由数据放本地缓存就可以了，我们看下图演示。



看到这里，我想大家对于Producer是如何拉取Topic路由数据的，就应该很清楚了，说白了底层主要就是基于Netty去发送网络请求而已，并没什么太难的东西，然后就是一些本地缓存更新的逻辑，大家有兴趣，可以自己去看看对应的源码，有了专栏中的思路讲解，你看懂源码应该就很容易了。

End

专栏版权归公众号狸猫技术窝所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

- [《从零开始带你成为JVM实战高手》](#)
- [《21天互联网Java进阶面试训练营》（分布式篇）](#)
- [《互联网Java工程师面试突击》（第1季）](#)
- [《互联网Java工程师面试突击》（第3季）](#)